Unifying recommendation and active learning for information filtering and recommender systems

Scott Cheng-Hsin Yang^{1*}, Chirag Rank¹, Jake Alden Whritner², Olfa Nasraoui³, and Patrick Shafto¹

¹Department of Mathematics and Computer Science, Rutgers University, Newark, NJ, 07102, USA

²Center for Perceptual Systems, University of Texas at Austin, Austin, TX, 78712, USA

³Department of Computer Engineering and Computer Science, University of Louisville, Louisville, KY, 40292, USA

*Corresponding author: Scott Cheng-Hsin Yang,
scott.cheng.hsin.yang@gmail.com

Abstract

The enormous scale of the available information and products on the Internet has necessitated the development of algorithms that intermediate between options and human users. These AI/machine learning algorithms attempt to provide the user with relevant information. In doing so, the algorithms may incur potential negative consequences stemming from the need to select items about which it is uncertain to increase predictive accuracy versus the need to select items about which it is certain to increase recommendation accuracy. This tension between predicting relevant recommendations to the users and learning about the user's interests can be considered an instantiation of the well-known exploration-exploitation tradeoff in the context of information filtering and recommender systems. Building from existing machine learning algorithms, we introduce a parameterized model that unifies and interpolates between recommending relevant information and active learning. We present three experiments investigating the unified model. Specifically, we illustrate the tradeoffs of optimizing prediction and recommendation within a tightly controlled concept-learning paradigm, show the

conditions under which a broad parameter range can optimize for both, and identify the effects of human variability on algorithm performance. Thus, combining methods and models from cognitive science and computer science, we quantify implications of tradeoffs between recommendation accuracy and learning about preferences of human users, demonstrating the value of experimental approaches to understanding real world human-machine feedback loops.

Keywords: Recommender system, information filtering, recommendation, active learning, exploration-exploitation tradeoff

1 Introduction

Historically, the information each individual had access to was defined by one's local environment: what one could directly observe, who one had to talk to or do business with, and available texts or catalogs one could access. With the advent of the Internet, information and products became available at a global scale. This vast potential resource creates a problem: how to choose—from billions or trillions of options—which information or products to present to an individual at a given time? Solutions to these problems form the foundation that supports major players in the online business world—from search engines and e-commerce to social network services—such as Google, Facebook, and Amazon (e.g., Linden et al., 2003). These algorithmic solutions radically affect not only what information and products we are exposed to, but also which information and products we have the chance to be exposed to. Thus, these algorithms mediate between us and reality, not by providing a random sample from what is possible, but by carefully selecting a sample which optimizes some underlying goals and metrics. The consequences of these human-algorithm interactions have been insufficiently explored despite recent interest in resulting problems such as filter bubbles (Pariser, 2011), algorithmic bias (Baeza-Yates, 2016), and human-algorithm interaction biases (Nasraoui and Shafto, 2016).

A well-established doctrine in cognitive science asserts that a driving factor of our beliefs is the information we are exposed to. However, the situations investigated in the most typical concept learning experiments (Bruner et al., 1956; Shepard et al., 1961) differ sharply from the kinds of situations we encounter with recommender systems. In concept learning experiments, examples are typically sampled either exhaustively or randomly, neither of which is feasible in the context of Internet-scale problems. Obviously, enumeration is not feasible. Random sampling is also not feasible because if the quality of the algorithm's suggestions were too poor, human users could simply choose to go elsewhere. This yields a thorny problem: how to select information and products to maximize relevance, while also

accurately estimating what users want.

Two classes of algorithms—information filters (Sparck Jones, 1970; Van Rijsbergen, 1979; Salton et al., 1983) and recommender systems (Goldberg et al., 1992; Maes et al., 1994; Adomavicius and Tuzhilin, 2005)—have been developed to facilitate selection of information for users. Although different in some ways, they share a core assumption that the goal is to deliver relevant information or products to humans. Given that these sorts of algorithms have raised concerns about not exposing people to the breadth of potentially relevant information (Szpektor et al., 2013), one might ask whether the data they obtain allow them to accurately estimate human users' preferences, and whether there are small adjustments that could be made to optimize for both recommendation and learning about the users' preferences.

One approach for obtaining optimally informative data is *active learning*, well known in both cognitive science (Nelson, 2005; Coenen et al., 2019) and computer science (MacKay, 1992; Settles, 2009). Active learning has been proposed both as a model for how humans search for information (Oaksford and Chater, 1994; Markant et al., 2016; Yang et al., 2016) and as an algorithm for how machines learn about the world (Lindley, 1956; Sacks et al., 1989; Cully et al., 2015; Yang et al., 2017). In the current context, active learning is a method for learning what information is relevant to the human user (Elahi et al., 2016), which—while having advantages for estimating the user's beliefs—is unlikely to produce quality recommendations.

This tension between predicting relevant recommendations to the users and learning about the user's interests can be considered a form of the exploration-exploitation tradeoff. Recommendation is akin to exploitation in that it favors items similar to those that are known to have high ratings. Active learning is akin to exploration and is more query-efficient than random sampling. Studies that cast recommender systems in this exploration-exploitation framework mostly aim to balance the tradeoff by leveraging methods introduced in the reinforcement learning literature (Bouneffouf et al., 2012; Szpektor et al., 2013; Wang et al., 2014; Zhang et al., 2015; Liu et al., 2018; McInerney et al., 2018; Chen et al., 2019). A smaller subset of studies focus on investigating the learning rate of exploration methods (Boutilier et al., 2003; Houlsby et al., 2012) and explicitly characterizing both sides of the recommendation/learning tradeoff (Balabanović, 1998; Herlocker et al., 2017). Most of these studies evaluate their methods using offline log data; however, the sparseness and static nature of log data force the evaluation to be conducted on limited and suboptimal items. Several companies have conducted online experiments to alleviate these problems, although they cannot afford to test exploration to the desired extent due to business concerns (Szpektor et al., 2013; McInerney et al., 2018; Chen et al., 2019).

In this work, we investigate the range of the exploration-exploitation tradeoff compu-

tationally and behaviorally. We begin by providing a unifying framework that *interpolates* between recommendation and active learning. In contrast to popular methods that address the exploration-exploitation tradeoff in reinforcement learning, this interpolation is novel because it is not a mixture of strategies for balancing the tradeoff and because its spectrum covers exploitation and active exploration as opposed to exploitation and random exploration. We then conduct a series of three tightly controlled human experiments where we vary the degree of calibration on prior knowledge as well as the complexity of the recommendation domain. We characterize the tradeoff between recommendation to users and learning about the users along the proposed interpolation in both conventional laboratory settings and a real-world recommender system platform. Through this detailed characterization, we identify situations in which the tradeoff is almost non-existent. We also show that human variability can strongly influence the algorithm's recommendation quality and predictive power. Overall, the main contributions of our work are: (i) the introduction of a unifying framework for recommendation and active learning, (ii) the use of an experiment to gauge how real human users interact with algorithmic systems that span graded shades between recommendation and active learning, (iii) conducting tightly controlled human experiments in the real-world domain of music recommendation, and (iv) the use of the unifying framework to interpret and characterize the exploration-exploitation behavior of information filtering and recommender systems in the face of naturalistic human variability.

2 Unifying recommending and active learning

In this section we introduce the theoretical framework that interpolates between pure recommendation and active learning. The framework takes as input a probabilistic classifier that makes prediction on each item in the inventory or stimulus pool. Then, a selection algorithm chooses which item to show to the user based on the predictions of the classifier. The purpose of the selection algorithm could be purely for recommendation, purely for active learning, or an interpolation between those two goals.

Given a training dataset, $D = \{x_i, y_i\}_{i=1}^N$, a probabilistic classifier can predict the probability that a new example x^* belongs to class y, $P(y|x^*, D)$. We are concerned with learning two classes corresponding to relevant (y = 1) and irrelevant (y = 0) information. In the context of recommender system, relevant and irrelevant information map to like and dislike responses, respectively. These predictions form the basis of the recommendation and active learning algorithms we used in the experiments. Intuitively, the goal of recommendation is to provide the user with examples or items that are relevant. This intuition can be formalized

directly by selecting item x_{rec} satisfying

$$x_{rec} = \underset{x^*}{\arg\max} P(y = 1 | x^*, D).$$
 (1)

Given previously observed data, this equation defines which item is optimal for recommendation: the one that maximizes the probability of being relevant. Within the closely related problem of probabilistic retrieval (ranking relevant information), this coincides with optimal probabilistic retrieval for the most relevant item (Robertson, 1977).

One intuitive formalization of active learning is to select examples that reduce the uncertainty about the relevance of the examples. This can also be formalized directly by selecting item x_{act} satisfying

$$x_{act} = \underset{x^*}{\arg\min} |P_{\text{median}} - P(y = 1|x^*, D)|,$$
 (2)

where P_{median} is the median of the distribution of the predictive probabilities induced by the classifier on the stimulus pool, $\{x_j^*\}_{j=1}^M$. Again, given previously observed data, this equation defines the optimal example for active learning: the one about which we have the greatest predictive uncertainty. Operationally, maximum predictive uncertainty is reached when the predicted probability coincides with P_{median} . In the first two experiments, the stimulus pool induces a uniform distribution over the entire range of predictive probability values. This implies that $P_{\text{median}} = 0.5$. In the third experiment, which is a more general scenario, P_{median} changes dynamically and is determined by the median of the empirical distribution of predictive probabilities.

It is worth noting that this is not the only formalization of active learning that one may consider. Other well-known strategies include optimizing information gain (K-L divergence), diagnosticity, and probability gain (Nelson, 2005; Crupi et al., 2018). While each differs in formal detail, in many practically relevant situations, their predictions are similar. We formalize active learning by Equation 2 to facilitate integration with the recommendation criterion in Equation 1, as shown below.

We propose a unified model of active recommendation that exploits the parallel structure in the previous two models. Our single parametrized generalization includes pure recommendation and active learning as extreme cases and thus unifies the two approaches and interpolates between them via parameter α . Formally,

$$x_{\alpha} = \underset{x^*}{\arg\min} |P_{\alpha} - P(y = 1|x^*, D)|,$$
 (3)

where P_{α} indicates a particular $(100\alpha)^{th}$ percentile of the distribution of predictive proba-

bilities induced on the pool of new examples. For example, when $\alpha = 0.5$, P_{α} corresponds to the 50th percentile of the distribution, which means $P_{0.5} = P_{\text{median}}$. In this case we recover active learning, as is obvious from inspection of Equation 2. When $\alpha = 1$, we recover pure recommendation, as subtracting from $P_1 = P_{\text{max}}$ and taking the minimum in Equation 3 is equivalent to taking the maximum in Equation 1.

In Experiments 1 and 2, we explore the range $\alpha = [0.5, 1]$ and study what happens between these extremes that correspond to recommendation ($\alpha = 1$) and active learning ($\alpha = 0.5$). In particular, are there parameterizations of active recommendation that produce both high recommendation and prediction accuracy? We also investigate the robustness of the parameterizations to the kinds of variability that are characteristic of human decision making. In Experiment 3, we explore the full range $\alpha = [0, 1]$, where $\alpha = 0$ can be considered anti-recommendation. Although this anti-recommendation case has no real-world usage, it provides better statistical power for teasing out certain expected trends and symmetries (see Section 3.3 for details).

3 Experiments

We introduce a novel experimental framework that merges the real-world problem of information filtering with experiments that are typical in concept learning. This framework allows for an exploration of selection algorithms that perform better or worse on the joint problems of optimizing recommendations and inferring relevance. The framework consists of three phases: a calibration phase where people are trained on a pre-defined relevant concept, a recommendation phase where people interact with the selection algorithm, and a test phase where people respond to items designed to uniformly cover the stimulus space (Fig. 1). In the recommendation phase, the classifier is updated each time the user responds to a newly recommended item, and the classifier is used to predict the user's responses at the different time points in the test phase. Note that the selection algorithm not only determines the quality of recommendation in the recommendation phase; however, it also affects the classifier's prediction accuracy on the responses in the test phase because the updating of the classifier depends on the items selected in the recommendation phase. Thus, the quality of recommendation and prediction derived from the two phases can be used together to probe the exploration-exploitation tradeoff in the context of information filtering and recommender systems.

Three experiments follow. Experiment 1 investigates the performance of pure recommendation and active learning and compares them with a random selection algorithm. This experiment allows us to validate that recommendation ($\alpha = 1$) and active learning ($\alpha = 0.5$)

fail to predict and fail to recommend well, respectively, and provides a random-sampling baseline. Experiment 2 investigates the unified active recommendation model which interpolates between pure recommendation and pure active learning. This experiment characterizes the behavior of recommendation and prediction accuracy along the interpolation, focusing on whether the tradeoff between the two types of accuracy would remain intact, worsen, or be ameliorated. These first two experiments are seated in a simple concept-learning task. Experiment 3 investigates the same questions studied in Experiments 1 and 2 about active recommendation, but extends the domain to the real-world domain of music recommendation. The calibration phase is omitted in Experiment 3 to keep the natural environment of information-filtering and recommender systems.

The three experiments combined offer a unique set of features that makes our design superior to popular existing methods for the study of human-machine learning algorithm feedback loops. These features are: (1) well-defined and calibrated ground truth of the user's relevance landscape, (2) naturalistic human variability in responses, (3) naturalistic complexity in relevance landscape, and (4) the freedom to test any item. Although the first and third features cannot co-exist in a single experiment, the closeness in design among the three experiments allows us to connect their findings. In contrast, existing methods fall short of these features (e.g., Balabanović, 1998; Boutilier et al., 2003; Bouneffouf et al., 2012; Szpektor et al., 2013; McInerney et al., 2018; Chen et al., 2019). For example, computational simulations do not include human variability, and neither offline analysis of log data nor online user testing have access to the ground truth or the freedom to test any item.

3.1 Experiment 1: Active learning versus recommendation

Experiment 1 is situated in a simple concept-learning task where the concepts, or stimuli, vary along two feature dimensions (Markant and Gureckis, 2014). Participants were first trained on the underlying conceptual structure that defines which examples are relevant and which are not. The classes of relevant examples are defined by axis-aligned logistic function with data standardization in two dimensions. Next, people are randomly assigned to a selection algorithm (recommend, active learning, or random), and are presented with a series of examples, which they label as being in the relevant class or not. The classifier updates itself upon receiving each example-label pair and then outputs its predictions to the selection algorithm for the selection of a new example.

The main purpose of this experiment is to validate our approach by demonstrating the expected limitations of pure recommendation and active learning: the tradeoff that pure recommendation would lead to high recommendation accuracy but low predictive accuracy

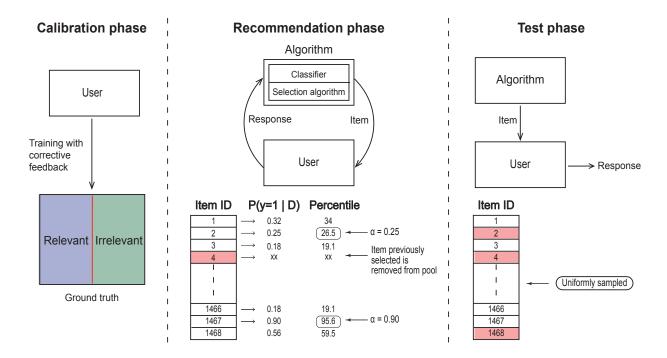


Figure 1: The three phases of the experiment. Experiments 1 and 2 include all three phases; Experiment 3 omits the calibration phase. Calibration phase: The user's relevance landscape is calibrated to match a pre-specified decision boundary (ground truth) through training with corrective feedback. **Recommendation phase:** The selection algorithm, parameterized by α , selects an item and shows it to the user. The user provides a response to the recommended item. The classifier updates itself with the newly received item-response pair and feeds new predictions to the selection algorithm for the selection of the next item. This loop continues until the end of the phase. The bottom part shows the selection process described by Equation 3. The classifier makes predictions on the relevance, i.e., P(y=1|D), of all items not previously recommended (non-red). Percentiles on the distribution of relevance are computed. The item with the percentile closest to the pre-defined parameterization α is selected as the item to recommend. **Test phase:** The algorithm does not incorporate user feedback. Responses are recorded for offline evaluation of the classifier. Items are selected to cover uniformly the stimulus space. Experiments 1 and 2 deterministically sample items from a grid. Experiment 3 samples uniformly randomly from all items not previously recommended, as depicted by the bottom part.

and vice versa for active learning. The selection algorithm's trial-by-trial recommendation accuracy is the fraction of examples labeled as relevant, at each trial, by a population of participants. The classifier's predictive accuracy for trial i is the fraction of correct predictions made by the classifier trained with data (from the recommendation phase) up to trial i. Here, the predictions are tested on a grid of predetermined, held-out test examples. The correctness of a prediction is judged against the optimal decision boundary that was set in the beginning of the experiment.

3.1.1 Methods

Participants. The experiment was run on Amazon's Mechanical Turk (MTurk) with 30 participants in each of the three conditions: recommend, active learning, and random.

Stimuli. Following Markant and Gureckis (2014), the stimuli were circles with a central diameter. The stimuli varied along two dimensions—the size of the circle's radius in pixels and the orientation of the central diameter in degrees (for an example, see Figure 5B in Markant and Gureckis (2014)). The ranges of the size and orientation were fixed to 110 pixels and 140 degrees, respectively. The minimum radius and minimum orientation for the classes were sampled independently and uniformly from 10 to 30 units and fixed for the whole experiment. This procedure determined a pair of minimum and maximum values $\{min, max\}$ for each dimension.

For each instance of the experiment, one of the dimensions (size or orientation) was randomly selected as the separable dimension. Let $\{min_s, max_s\}$ be the minimum and maximum values of the separable dimension, and $\{min_t, max_t\}$ be the values for the other dimension. Two classes were defined by two two-dimensional normal distributions. Along the separable dimension, the variances of the two classes were both 75, and their means were set at $(min_s + max_s)/2 \pm 30$. A linear decision boundary at $(min_s + max_s)/2$ is set to be the ground truth. Along the other dimension, the variances were both 2250, and the means were both $(min_t + max_t)/2$. Stimuli were sampled from the two-dimensional Gaussian described above. Those that happened to be outside the determined range were resampled.

The experiment consisted of three phases: calibration, recommendation, and test. In each trial of the calibration phase, a class was randomly sampled, and a stimulus was sampled according to its class distribution. In the recommendation phase, there were three selection or sampling algorithms. First, random sampling used the same procedure as in the calibration phase. Recommendation and active-learning sampling followed Eqs. 1 and 2, respectively. The selection was made from a fixed pool of 400 randomly sampled stimuli for each instance of the experiment. In the test phase, the stimuli were no longer sampled from the classes but from a test set. The test set consisted of 16×16 samples that lied on a regular grid covering the area of the feature space defined by [10,140] pixels \times [10,170] degrees. Five stimuli were randomly selected from each of the four quadrants in that area to form the 20 test stimuli used in the test phase.

Procedures. Before the calibration phase, participants were instructed that throughout the experiment, they would see a series of "loop antennas" that receive signals from music stations called "Beat" and "Sonic" (the two classes of stimuli described above). They were instructed that the station received depends upon the antenna's radius and the orientation of its diameter. The goal of the calibration phase, as described to the participants, was for them

to learn which station was received by a given class of antennas (e.g., Beat antennas have large diameters and Sonic antennas have small ones). Participants provided input by clicking on one of two buttons labeled Beat and Sonic, respectively. After responding, participants received feedback on whether or not their input was correct. The participants moved on to the recommendation phase once they had 19 correct answers in the past 20 trials.

The recommendation phase was comprised of two parts. Participants were first instructed to pretend that they preferred either Beat or Sonic. Given this preference, participants were told that they would teach an algorithm to recommend the station that they preferred by indicating that the antenna it chose was one that they liked or disliked through clicking one of the two buttons labeled "like" and "dislike." Participants were instructed to pay attention to whether the algorithm was improving or not. This part of the recommendation phase continued for 20 trials. Next, participants rated the algorithm's improvement—that is, how well the participants thought the algorithm learned to recommend their preferred station—using a slide bar from "very poor" to "excellent."

The final phase of the experiment, the test phase, entailed a classification test to confirm whether participants still remembered the classes correctly. This phase followed the same procedure as the initial calibration phase, but did not provide participants with feedback (i.e., they were not told whether or not their classification was correct). Afterwards, participants were asked to provide feedback about the experiment and identify the rule behind the classification they were trained on. Participants who successfully completed all phases of the experiment were compensated via MTurk.

3.1.2 Results

We quantify the behavior of the algorithms (the classifier with the different selection algorithms) by their trial-by-trial recommendation and predictive accuracy, as described previously. We report the first trial index by which an algorithm's recommendation accuracy becomes statistically different from 50% as well as the first trial index at which its recommendation accuracy becomes statistically no different from 95%. For these we use the binomial test and claim statistical significance when the p-value is less than 0.05. We also report the trial index at which the classifier's predictive accuracy converges. We formalize this as the first trial at which the predictive accuracy is not statistically different from the prediction accuracy at the last trial, using a one-sample t-test. The accuracy at the final trial is reported as the converged value.

We omit participants whose test accuracy is below 18 out of 20 (below 90%), where test accuracy is defined by the fraction of station choices that match the ground truth. For the included participants, we compute a consistency score, which is the fraction of their responses

to the recommended examples that matched the expected responses. For participants whose consistency score is below 50%, we compute the predictive accuracy after flipping all their responses in the recommendation phase. This allows us to correct for the responses from participants who misremembered the preference during the recommendation phase. ¹ The recommendation accuracy is likewise computed after the flipping of responses. The number of included participants is 26/30 (3 flipped) for random, 27/30 (4 flipped) for active learning, and 27/30 (3 flipped) for recommendation.

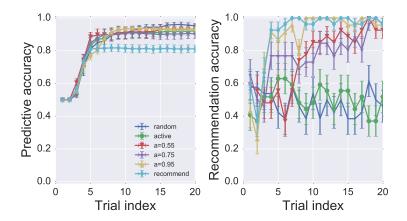


Figure 2: Predictive and recommendation accuracy for all six conditions as a function of trial index. The six conditions include conditions from both Experiments 1 and 2. The active-learning condition ($\alpha = 0.5$) is labeled "active," and the pure recommendation condition ($\alpha = 1$) is labeled "recommend."

Fig. 2 shows the recommendation and predictive accuracy of the different selection algorithms. As expected, examples chosen under the recommendation objective result in high recommendation accuracy, but low predictive accuracy. As a function of the number of examples seen (trial index), recommendation accuracy rises above chance level and reaches 95% after 4 examples, ² while predictive accuracy converges after 5 examples to 81%, which is low compared to the active-learning or random algorithms. Conversely, the active-learning algorithm leads to low recommendation accuracy but high predictive accuracy. As a function of trial index, recommendation accuracy remains at chance level, while predictive accuracy converges after 5 examples to 92%. For reference, results of the random algorithm are also presented. These show a pattern similar to that observed for active learning. There is a rapid increase in predictive accuracy, converging after 8 examples to 95%. Recommendation

¹The preference refers to which station the participants were asked to like and not the feature of the stations they learned in the calibration phase; thus, it is possible for the participants to have low consistency but high test accuracy.

²This and subsequent numbers reported in the Results sections of Experiment 1 and 2 represent statistically significant results, as described above.

accuracy remains at chance level throughout.

3.2 Experiment 2: Active recommendation

An ideal algorithm would have both high recommendation accuracy and high predictive accuracy. As a function of the number of examples given, one hopes that the recommendation accuracy will approach 1 after a few examples and that the predictive accuracy will steadily increase to 1. Given the sharp dichotomy between the performance on pure recommendation and active learning, it is not obvious how best to achieve this.

As mentioned in the Introduction, most existing works aim to achieve this by balancing exploitation (recommendation) and exploration (active learning). Here we explore a one-parameter generalization of recommendation and active learning that interpolates between the two (see Section 2). The difference between balancing and interpolation is that the former is a strategy that shifts between the two modes, while the latter generates novel modes.

This generalization, active recommendation, provides selection algorithms parameterized by α , with $\alpha = 1$ and $\alpha = 0.5$ producing pure recommendation and active learning, respectively. In Experiment 2, we study the parameterizations of $\alpha = \{0.55, 0.75, 0.95\}$ as interpolation points. We investigate the behavior of predictive and recommendation accuracy as a function of α . The questions of interest are: which selection algorithms (which α values) perform well in terms of recommendation and prediction accuracy and which ones perform well in terms of robustness to human variability?

3.2.1 Methods

The experiment was run on MTurk with 30 participants for each of the 3 conditions: $\alpha = \{0.55, 0.75, 0.95\}$. Following the criteria described in Experiment 1, the number of participants included in the analysis is 27/30 (4 flipped) for $\alpha = 0.55$, 26/30 (5 flipped) for $\alpha = 0.75$, and 23/30 (3 flipped) for $\alpha = 0.95$. The stimuli and procedures are the same as Experiment 1.

3.2.2 Results

In Fig. 2, the predictive accuracy in the active-recommendation conditions converge to 93%, 90%, and 93% for $\alpha = \{0.55, 0.75, 0.95\}$, respectively. These are similar to the 92% in the active-learning condition and better than the 81% in the recommend condition. The predictive accuracy of the active-recommendation conditions converged after

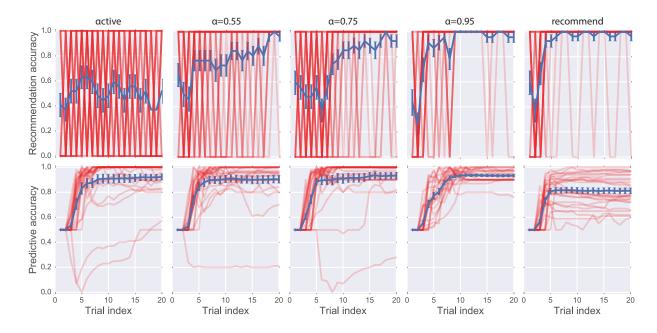


Figure 3: Recommendation and predictive accuracy for all included participants broken down by condition. Red traces correspond to individual participants, and the blue trace is the average. Note that a few mislabeled examples in the early trials can lead to unstable behavior, such as those curves that dip below chance level.

5, 5, and 8 examples for $\alpha = \{0.55, 0.75, 0.95\}$, respectively. The recommendation accuracy in the active-recommendation conditions reached 95% after 12, 16, and 4 examples for $\alpha = \{0.55, 0.75, 0.95\}$, respectively. These are similar to the recommend condition in that all reached 95%, whereas the recommendation accuracy in the active-learning and random conditions remain at chance level.

Importantly, if we move slightly away from the active-learning condition (sampling slightly away from the decision boundary by moving from $\alpha=0.5$ to 0.55), we can achieve much higher recommendation accuracy while also achieving similar predictive accuracy. The recommendation accuracy under the $\alpha=0.55$ condition rises above chance after 8 examples and reaches 95% after 12 examples, whereas that under the active-learning condition remains at chance level throughout. Similarly, if we move slightly away from the recommendation condition (from $\alpha=1$ to 0.95), we can maintain the recommendation accuracy while improving the predictive accuracy. Thus, these intermediate conditions of α appear to allow the algorithms to uncover more of the relevant space compared to pure recommendation.

Fig. 3 employs the same measures as Fig. 2, but displays each of the five α conditions separately, with individual participant performance (red lines) and the results averaged over all participants (blue lines). Fig. 3 allows for a closer look at individual variability during the experiment, and in particular highlights the difference in recommendation accuracy from

the active-learning and $\alpha=0.55$ conditions. Comparing the density of the red traces in these two conditions in Fig. 3, we can see that variation in recommendation accuracy across individuals persists for all trials in the active-learning condition, while reducing greatly after 8 to 12 trials in the $\alpha=0.55$ condition. Comparing the red traces in the $\alpha=0.95$ and recommend conditions, we can see the predictive accuracy across individuals varies much less in the $\alpha=0.95$ condition than in the recommend condition, resulting in the better average predictive accuracy for $\alpha=0.95$.

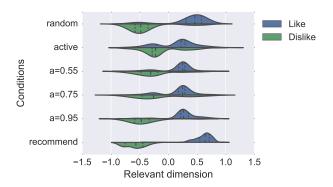


Figure 4: The distributions of like/dislike examples for each condition in the recommendation phase. The relevant dimension is the separable feature dimension in which the classes are defined, and the zero point marks the ground-truth decision boundary. The dotted lines in the distributions indicate the distributions' quartiles.

The cause of the improved performance of intermediate α values can be traced back to the examples the algorithms select. The distributions of "likes" and "dislikes" are plotted in Fig. 4 alongside random sampling, active learning, and pure recommendation. At the top, random sampling replicates the true distribution (up to some small number of inconsistent responses). The active-learning algorithm selects examples that are evenly distributed across likes and dislikes but shifted toward the decision boundary between the two classes. At the bottom, the pure recommendation algorithm selects examples that are skewed away from the boundary and the balance of examples is strongly tilted toward likes, consistent with the goal of recommending relevant examples. Of particular interest are the three activerecommendation conditions. There are minor differences focused on the distribution of disliked items. What is most notable is the similarity between the distribution for likes in the active-learning condition and those in the active-recommendation conditions. Unlike the recommend condition, the selection algorithms in all three intermediate conditions disproportionately select "liked" examples that are close to the decision boundary. They all also select relatively few "disliked" examples. Cross-referencing against Fig. 3, these disliked items appear only in the early trials. To summarize, the advantage of the activerecommendation approach is a bias to select uncertain items within the relevant class. This allows them to achieve both high recommendation and predictive accuracy.

Interestingly, if we include only the fully consistent participants, the α values dictate a strict ordering in both the predictive and recommendation accuracy. Increasing α from 0.5 to 1, one sees a monotonic decrease in the converged predictive accuracy and a monotonic increase in the rate at which recommendation accuracy reaches 1. The stochasticity in the participants' responses can break the ordering in two ways. First, selection algorithms that provide examples closer to the decision boundary will receive more noisily labeled examples. Second, randomness in responses slows down the convergence of the classifier. These effects cause the converged prediction accuracy, in conditions with smaller α values, to be lower than what they could be with less variable responses.

3.3 Experiment 3: Active Recommendation in the real-world domain of music recommendation

Our goal is to understand the tradeoff between recommendation and accurate estimation of beliefs or preferences. The first two experiments investigated an integrated approach to understanding recommendation and active learning. By tightly controlling people's beliefs, we showed that the generalized model can achieve high accuracy in estimating beliefs and high accuracy in recommendations. Our goal in this experiment is to test our model of active recommendation in the complex, real-world domain of music recommendation. In this experiment, participants underwent a test phase in which they were presented uniformly randomly sampled music, and they labeled as "like" or "dislike". The test phase was followed by a recommendation phase, in which the algorithm selected and played new music that the participants also labeled. Because the selection in the recommendation phase was not uniformly random, placing the recommendation phase before the test phase might cause some distributional and response biases in the test phase. Thus, the test phase was positioned before the recommendation phase, although we expect these biases to be negligible. As in the previous experiment, the goal is high recommendation accuracy in the recommendation phase and high prediction accuracy in the test phase.

The music was obtained from Spotify, and came with 11 features describing each song. We use these features to perform prediction and recommendation. The 1468 songs used spanned 123 genres including pop, classical and salsa. The classifier was a Gaussian Process (GP) classification model with a squared exponential kernel. During the recommendation phase, each labeling of a song as liked or disliked triggered retraining of the model, and the next song was selected to optimize an objective parameterized by α according to Equation 3. Notably, compared to Experiments 1 and 2, the construction, testing and optimization of

this system required a large amount of work. The open-sourced code for the system can be found on Github at https://github.com/CoDaS-Lab/MusicWebApp and https://github.com/CoDaS-Lab/MusicApi. An overview of the system architecture can be found in the README of the repositories.

One of the principal challenges in real-world recommendation is that the base rate of liking music (or anything!) varies across people. As a consequence of differing base rates, the participants' labelling may contain more or less information. For example, for an individual that likes most things, it is more informative to find out what they do not like (and vice versa). Because the classifier and selection algorithm are personalized based on each individual's labelling, an accurate assessment must capture differences across people and their consequences for recommendation and prediction. These differences are calibrated away in Experiments 1 and 2 in the calibration phase. However, in this experiment, we maintained the typical music recommendation setup without a calibration phase and addressed the differences by modelling the individuals' base rates and accounting for them in our statistical analysis of the recommendation and prediction accuracy. The set of α values we consider in this experiment is $\alpha = \{0, 0.25, 0.5, 0.75, 0.9, 1\}$. The condition of $\alpha < 0.5$ can be thought of as anti-recommendation: purposely selecting music that are relatively dislikable among the pool. These conditions have no practical usage, but offer statistical power because of certain symmetries we expect in the relationship between α and recommendation and prediction accuracy.

3.3.1 Methods

Participants. The participants were recruited from Amazon Mechanical Turk. The location of the participant was restricted to the United States. A total of 240 people participated in the experiment. The participants were randomly assigned (40 per condition) to one of the six conditions: $\alpha = \{0, 0.25, 0.5, 0.75, 0.9, 1\}$. All 240 participants were paid \$3 upon completing the experiment. Eleven participants were removed from the analysis because they provided the same label (either like or dislike) for the first 40 songs in the recommendation phase.

Stimuli. A total of 1468 songs covering 123 genres were obtained from the Spotify API. These songs covered all but 3 genres that Spotify offered at the time of retrieval. The three excluded genres were comedy, metal-misc, and synth-pop because they contained only speech. For each song, the API provided a music ID, a popularity score, 11 feature values, a list of associated genres, and a 30-second preview sample. The popularity score of this pool was roughly normally distributed. The 11 features used by the GP classifier were danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness,

liveness, valence, tempo (see Spotify API documentation³ for details). These features were standardized to have a population mean of 0 and a population standard deviation of 1. In the selection process, each song was assigned to a unique genre group by using the foremost genre in its list of associated genres. In presenting the songs, only the first 10 seconds of the preview samples were used.

Procedures. The experiment was divided into two phases: a test phase followed by a recommendation phase. The music were selected from the fixed pool of 1468 songs mentioned above. Sampling without replacement was used throughout the experiment to avoid repeated presentation of the same music. Music were randomly sampled in the test phase and selected according to Equation 3 in the recommendation phase.

The participants were informed that the experiment was divided into two phases but not of their difference. Each phase lasted for 7 minutes. The 7-minute counter counted only music playing time and did not include the participants' decision-making time after listening to the music, the algorithms' computing time, or the music loading time. In each trial of both phases, participants listened to a song presented and provided their response by clicking on a thumb-up or thumb-down icon for liking or disliking that song, respectively. They could provide their response for a song anytime during or after the 10-second playtime of the music. The participants also had the choice to pause the music anytime by clicking on the pause icon. Upon submission of the response, the playtime of the music at the time of the submission was recorded along with the response of the participant. Then, the next trial began immediately with the selection, loading, and playing of the next song.

The test phase proceeded as described above with random selection of songs for a total of 7 minutes of music playtime, which ensured at least 42 trials and led to an average of 74.2 trials (SD=37.4). In the recommendation phase, the selection of music was done according to Equation 3 and depended on the α condition the participant was assigned to. The GP classifier was implemented using the Scikit-learn package (Pedregosa et al., 2011) with the default setting, which used the radial-basis-function kernel (or the squared-exponential kernel) with hyperparameter optimization. ⁴ The default setting of the classifier required at least one label from each of the two classes; thus, random sampling was used for the selection of songs before this criteria was met. There were on average 74.7 trials (SD=42.3) in the recommendation phase. When the participants concluded the recommendation phase, they were shown a screen in which they were thanked and debriefed. The debriefing consisted of a free-form survey question asking for general feedback. All the participants that successfully

³https://developer.spotify.com/documentation/web-api/reference/tracks/ get-audio-features/

 $^{^4} https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process. \\ GaussianProcessClassifier.html$

completed the experiment were compensated via Mturk.

3.3.2 Results

Our goal here is to understand the relationship between recommendation and prediction for the different conditions parameterized by α . The results from Experiment 1 shows that there is a tradeoff between recommendation and prediction. The algorithm designed for active learning ($\alpha=0.5$) achieves good prediction but provides poor recommendation; the algorithm designed purely for recommendation ($\alpha=1$) provides desirable recommendation but loses predictive power. Here, we expect the same overall trend, but generalized to the range of $0 \le \alpha \le 1$: Extreme recommendations based on pure recommendation and pure anti-recommendation lead to poor prediction, while good prediction leads to mediocre recommendation relative to the two extreme recommendations. The results from Experiment 2 show that both recommendation and prediction can be made effective in the active-recommendation conditions ($0.5 < \alpha < 1$). This feature of active recommendation hinges on the algorithm's ability to fully capture the user's choice landscape. Since this is difficult to achieve for the current experiment, where the users' choice landscapes are high-dimensional and complex compared to the amount of data received, we focus on analyzing the variation induced by changes in α rather than the algorithms' raw performance.

We investigate the tradeoff by studying the recommendation accuracy and the predictive probability both as a function of α . Recommendation accuracy is defined to be the fraction of "like" responses in the recommendation phase. This fraction can be an average over the trials for a participant or an average over the participants for a given trial. Note that the selection algorithm began making recommendation only after a user's responses included one "like" and one "dislike" choice; thus, the trial on which the first recommendation took place varied from participant to participant. For the average over trials, responses before the first recommendation are left out of the average. For the average over participants, a constant, participant-specific padding value called base rate is assigned to all the trials before the first recommendation. The base rate of a participant is the fraction of "like" responses that the participant made in the test phase; it is an estimate of a participant's inherent bias towards "like" responses without any influence from the algorithm. The predictive probability for a participant on trial i is the average of the probabilities that the classifier assigns to all the participant's responses in the test phase after being trained on the participant's i responses in the recommendation phase. Predictive probability is used in place of predictive accuracy because the latter, being an average of binary values, is more susceptible to fluctuations than the former, which is an average of probabilities. Similar to recommendation accuracy, predictive probability can also be averaged over trials for a participant or over participants for a trial. For the average over trials, trials before the first recommendation are not included since the classifier cannot make predictions for these trials. For the average over participants, a padding value of 0.5 is assigned to the trials before the first recommendation across all participants. This value represents the probability of the classifier's correct prediction before receiving any training data.

3.3.3 Recommendation accuracy

We begin by studying the relationship between recommendation accuracy and the α parameter, which we expect to be monotonic. We first observe that in contrast to the previous experiments, participants' choices of "like" and "dislike" were not calibrated to a known, controlled standard. This is evident from the variation in base rates (min=0, Q1=0.234, median=0.344, Q3=0.525, max=1). Plotting base rate against recommendation accuracy (Fig. 5, left panel), we see that the base rate is a dominant predictor of the recommendation accuracy across the α conditions ($r=0.812, p<10^{-54}$). This points out the obvious: a participant with an extreme base rate will, with high probability, like or dislike almost any music. Thus, it is important to account for this inherent bias to more accurately surface the influences of the algorithm. In other words, the base rate can be seen as a base-line model for decision making in the recommendation phase, and the effect of the selection algorithms' recommendations are measured relative to this base-line model.

The middle panel in Fig. 5 shows the recommendation accuracy, after subtracting away the base rate, versus the α parameters. This difference is computed for each participant (averaged over trials) and represents the degree to which the selection algorithm shifted the participant's probability of making a "like" choice from his or her base rate. A simple linear model (difference $\sim \alpha$) captures the relationship between this adjusted recommendation accuracy and the α parameter ($R^2=0.197$). The positive slope of 0.186 ± 0.025 ($t(227)=7.463,\ p<10^{-11}$) and the negative intercept of -0.119 ± 0.017 ($t(227)=-7.122,\ p<10^{-10}$) are both significant, suggesting that the selection algorithms shifted the participants' choice behavior in the expected direction to the expected ordering. A bootstrap analysis with 10^5 fits to data sampled with replacement results in a zero-crossing of $\alpha=0.643\pm0.051$ (95% CI [0.563, 0.729]). Note that the active learning condition (α =0.5) is outside of the confidence bound, and the difference $\Delta\alpha$ between the zero crossing and this condition is approximately 0.14. We will explain this observation together with the predictive probability in the next section.

The right panel in Fig. 5 shows the same trend in the trial-by-trial recommendation accuracy averaged over participants. The y values are computed by (1) subtracting the base rate from the binary response of each participant, (2) averaging the difference over the

participants for each trial, and (3) performing a cumulative moving average along the trial number. The x-axis is the trial number in the recommendation phase, starting with the first song presented (instead of the first recommendation). The resulting curves all begin at zero because of the base-rate padding used for the trials before the first recommendation. After some fluctuations in the first 10 trials or so, the curves exhibit smooth trends and are well separated in the expected ordering (except for the $\alpha=0.75$ and 0.9 conditions) by around trial 30. The trends can be categorized into three distinct groups: the highest α condition ($\alpha=1$), the two lowest α conditions ($\alpha=\{0.5,0.75,0.9\}$) lead to an increasing, decreasing, and constant trend, respectively. Note that because the experiment controlled for the total time instead of the number of trials, each participant responded to a different number of songs. There is a large drop in number of participants around trial 47, and only about 15 participants remained in each condition by trial 60. The remaining trials are thus not shown because of the decreasing reliability.

In summary, these results align with the expected effect of recommendation: a selection algorithm designed to provide relatively likable or dislikable recommendations increases or decreases the participants' tendency to like those recommendations, respectively, whereas a selection algorithm aimed at probing the user's decision boundary ($\alpha \approx 0.5$) does not change the participants' choice behavior from their base rate much.

3.3.4 Predictive probability

Next, we study the relationship between predictive probability and the condition parameter α . Our expectation is that predictive probability will be lowest in conditions with extreme α values and highest around the active learning condition ($\alpha \approx 0.5$). Similar to what was observed for recommendation accuracy, base rate greatly influences the predictive probability (Fig. 6, left panel). This observation is also intuitive: it is easier to predict the responses of participants who have extreme base rates, as predicting "like" or "dislike" for any song would already achieve high predictive accuracy and probability. Inspired by this rationale, we construct a base-line prediction model called the *mean-field oracle*. The mean-field oracle model assigns the probability of "like" to be equal to the base rate for every song. Compared to the GP classifier used, this base-line model is an oracle because the classier does not have access to the base rate, and it is a mean-field model because it makes predictions based on an average probability regardless of the features of the song.

The predictive probability of the mean-field oracle model can be derived as follows. Let the base rate be denoted by b. In the test phase, the mean-field oracle would assign songs that are liked a probability of b, and songs that are disliked a probability of 1-b. Averaging

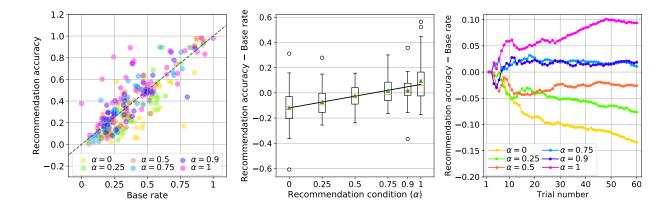


Figure 5: Recommendation accuracy. (Left). Recommendation accuracy averaged over trials versus base rate. Each point represents a participant. The strong correlation (r=0.812) shows that base rate strongly influences the recommendation accuracy. The y=x line shown represents the base-line model in which "like" choices are made with probability equal to the base rate. (Middle). Recommendation accuracy averaged over trials minus the base rate versus the recommendation condition parameterized by α . The boxplot indexed by α_i summarizes the participant-wise distribution for the α_i condition. The green triangles indicate the means of the distributions; the orange bars indicate the medians; and the black line is the linear fit $(-0.119 + 0.186\alpha)$ to all the data points. The negative intercept and positive slope of the fit suggest that the different recommendations shifted the participants' choice behavior in the expected manner. (Right). The difference between recommendation accuracy and base rate, averaged over participants, versus trial number. A marker shows the cumulative moving average of the average difference up to its corresponding trial number. The separation of the curves by α conditions reiterates the effect of the selection algorithm on recommendation accuracy on a trial-by-trial level.

over these probabilities, we have $[n_{\text{like}}b + n_{\text{dislike}}(1-b)]/n$, where n_{like} , n_{dislike} , and n denote the number of liked songs, the number of disliked songs, and the total number of songs presented in the test phase, respectively. Noticing that $n_{\text{like}}/n = b$ and $n_{\text{like}} + n_{\text{dislike}} = n$, we can express this average, which corresponds to the predictive probability of the mean-field oracle model, as $b^2 + (1-b)^2$. The left panel in Fig. 6 shows that this model captures the first-order relationship between predictive probability and base rate well $(r = 0.844, p < 10^{-62})$.

The middle panel in Fig. 6 shows each participant's predictive probability, averaged over trials, minus the corresponding prediction from the mean-field oracle model as a function of α . The trend of the difference takes an inverted U shape and can be captured by a linear model with both linear and quadratic terms in α (difference $\alpha + \alpha^2$; $R^2 = 0.051$). The significant negative coefficient on the quadratic term ($\beta = -0.107 \pm 0.041$, t(226) = -2.599, p = 0.010) confirms the concavity of the trend, which is consistent with the expectation that

the classifier's predictive power is lowest in conditions with extreme α values. A bootstrap analysis with 10^5 fits to data sampled with replacement suggests that the peak of the parabola is at 0.376 ± 0.080 (95% CI [0.221, 0.476]). Similar to the observation for recommendation accuracy, here the active learning condition is outside of the confidence interval. To explain this, we note that the peak value is very close to the average base rate across the participants, $\bar{b} = 0.379$. We suspect this is because on average, the classifier's predictions of the songs before sufficient training data are approximately unbiased at 0.5. In this case, it is the selection algorithm with $\alpha = \bar{b}$, and not $\alpha = 0.5$, that selects items closer to the true P_{median} of the participants' choice landscape. The average base rate also offers an explanation to the offset of $\Delta \alpha \approx 0.14$ in the zero-crossing observed in the analysis of the recommendation accuracy. Note that this $\Delta \alpha \approx 0.14$ is similar to $0.5 - \bar{b} = 0.12$. We suspect this is because on average, the neutral items selected in the active learning condition would be liked with the average base rate. Thus, it is $\alpha = 0.5 + (0.5 - \bar{b})$ that would be the condition with roughly equal number of "like" and "dislike" responses.

The right panel in Fig. 6 reiterates the ordering of the predictive probabilities in the different α conditions and reveals additional trial-by-trial behavior. The y values are computed by first subtracting the mean-field oracle's prediction from each participant's trial-by-trial predictive probabilities and then averaging over the participants for each trial. The trial number indicates the number of songs presented to the participants in the recommendation phase (same as that in Fig. 6, right panel). The upward trend of all the curves indicates that the classifier's prediction is improving as it receives more data in all the conditions. Negative values are expected in the beginning because the classifier has not received much data and should perform worse than the oracle model derived from test phase data. The classifier remains inferior to the mean-field oracle model in conditions with the two highest α values, achieves the same predictive power by around trial 50 in the $\alpha = \{0, 0.5, 0.75\}$ conditions, and clearly surpasses the oracle model after around trial 40 in the $\alpha = 0.25$ condition. Note that the ordering of the curves in the beginning is different from that at trial 60. The former ordering is a consequence of the variation in the participants' base rates, since the predictive probability is assigned a constant padding of 0.5 for all participants. In contrast, the latter ordering, which is consistent with that shown in the middle panel in Fig. 6, is a consequence of the influence of the recommendations provided in the different α conditions.

In summary, these results confirm that the classifier's predictive power increases in all conditions as training data increases. The extent of the increase is consistent with our expectation: recommendations aimed to exploit certainty in user responses lead to poor predictive power, whereas those aimed to resolve uncertainty in user responses are more helpful for training the classifier.

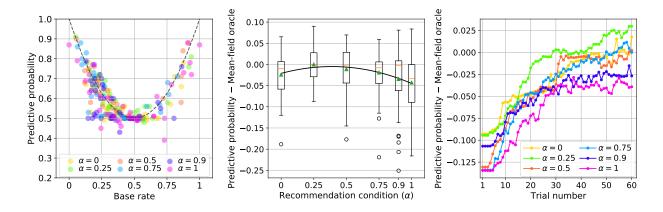


Figure 6: Predictive probability. (**Left**). Predictive probability averaged over trials as a function of base rate. Each point represents a participant. The dotted line represents the predictive probability of the mean-field oracle model. The correlation between the predictive probability of the participants and that of the mean-field model is 0.844. (**Middle**). The Predictive probability averaged over trials minus the predictive probability of the mean-field oracle model as a function of the parameter α that parameterizes the recommendation conditions. The boxplots use the same convention as in Fig. 5, middle panel. The black curve shows the least-square quadratic fit $(-0.172 - 0.107(\alpha - 0.389)^2)$ to all the data points. The concave trend indicates that the responses in conditions with extreme α values lead to poor predictive power, as expected. (**Right**). The predictive probability minus the output of the mean-field oracle model averaged over participants for each trial. The overall increasing trend of all the traces indicates that the classifier's predictive power is increasing as it receives more training data. The ordering of the traces after trial 40 is consistent with the concave trend shown in the middle panel.

3.3.5 The tradeoff between recommendation and prediction

Combining the middle panels in Figures 5 and 6, we observe that recommendations aimed at maximizing or minimizing a user's choice probability lead to lower predictive probability than recommendations aimed at maximizing uncertainty in user's choice probability. This recapitulates the basic tradeoff between active learning and pure recommendation shown in Experiment 1 in a real-world domain. From the perspective of the training of the classifier, the recommendations aimed at resolving uncertain choices maximally resolve the ambiguity about the decision boundary and thus lead to more informative training data. On the other hand, the recommendations aimed at eliciting certainty in users' choices avoid the vicinity of the decision boundary and thus lead to redundant training data.

A contrast between Experiment 3 and Experiment 1 shows that this basic tradeoff between recommendation and prediction is a robust phenomenon. Experiment 3 is more complex than Experiment 1 in at least three aspects. First, unlike Experiment 1, Experiment 3 does not enjoy the controlled calibration of users' decision rule to a known decision boundary.

This resembles most real-world scenarios where such calibration is unnatural and infeasible. The main consequence of the lack of calibration is the dominant effects of the base rate that we have observed. Second, compared to the low-dimensional linear decision boundary employed in Experiments 1 and 2, the decision boundary considered in Experiment 3 is nonlinear and the decision space is high-dimensional. This is true for both the users' actual decision process and the GP classifier used to model the users. The main consequence of this increased complexity is that the recommendations and associated responses are considerably less efficient in pinning down the true decision boundary. Lastly, an investigation of the algorithm's performance on individuals reveals large variations in the stimuli's coverage of choice probability (see Supplementary Materials). For some participants, the 1468 songs in the stimuli pool cover almost the entire range of [0, 1]; for many others, only a narrow range is covered. The coverage also changes with trial number, typically abruptly from wide to very narrow or vice versa. The appearance of the tradeoff despite these complications suggests its robustness and prominence.

The tradeoff observed in Experiment 3 does not capture the effect of active recommendation observed in Experiment 2. That is, the conditions with $0.5 < \alpha < 1$ in Experiment 3 do not show both high recommendation accuracy and high predictive probability. The core reason lies in the limitation of the classifier's predictive power. In the right panel in Fig. 6, the highest value the curves attain is around 0.025; however, the highest value a curve could potentially attain is roughly one minus the prediction of the mean-field oracle model for the average participant, which is $1 - [\bar{b}^2 + (1 - \bar{b})^2] = 0.479$, with $\bar{b} = 0.379$ being the participants' average base rate. This suggests that the classifier's predictive power in Experiment 3 is far from saturation. Active recommendation achieves good prediction and recommendation by probing choice uncertainty *inside* the relevant regions. This requires a fine enough characterization of the decision boundary that the classifier in Experiment 3 did not achieve yet. In other words, while the tradeoff is a robust phenomenon, the desired effect of active recommendation is a more intricate phenomenon that requires the classifier to be able to reach a sufficiently high prediction accuracy in the allowed time frame (trial numbers).

4 Discussion

Information filters and recommender systems mediate between humans and the vast information and product stores on the Internet. Naturally, these algorithms aim to provide relevant information, but this goal may also lead to negative consequences by overly restricting experience. Embedding recommendation into a concept-learning framework, ⁵ we investigated the algorithm's recommendation accuracy and predictive accuracy/probability in the presence of naturalistic human variability. We introduced a unified model called active recommendation, a family of algorithms parameterized by one parameter, that interpolates between pure (anti-)recommendation and active learning. We investigated the behavior of these algorithms in a series of three well-controlled experiments that spanned both conventional laboratory tasks and the real-world domain of music recommendation.

Our results show the expected tradeoff between recommendation and prediction in pure recommendation and active learning: Pure recommendation obtains high recommendation accuracy but low predictive accuracy/probability, while active learning leads to the reverse. Experiment 3 shows that this tradeoff can be observed in the real-world domain of music recommendation after accounting for the users' base rate—their inherent bias to liking or disliking music in general. This result presents a quantitative form of exploration-exploitation tradeoff in the domain of information filtering and recommender systems. When the classifier used for predicting user behavior falls short in performance in the allowed time frame, we observe a linear trend in the recommendation accuracy from pure (anti-)recommendation to active learning and a concave quadratic trend in predictive probability from pure antirecommendation to active learning to pure recommendation. These trends suggest that the tradeoff persists throughout the full spectrum of active recommendation. An interesting side discovery is that in the face of variable base rates, the most effective (non-participantspecific) active-learning algorithm is the active-recommendation algorithm that matches the average base rate $(\alpha = \bar{b})$ and not the nominal one $(\alpha = 0.5)$. When the classifier is data-efficient, we show that—across a wide range of parameterizations—active recommendation converges toward optimal predictive and recommendation accuracy. We also observe that parameterizations closer to pure recommendation yield better performance in terms of faster convergence and greater robustness to human variability in decision making. We trace the success of active recommendation to the fact that all parameterizations automatically combine rapid convergence toward selecting only relevant items and actively exploring informative examples from within that set. Parameterizations close to pure recommendation perform best because they minimize exploration of regions of the space where human actions are most variable—near the boundary and in the non-focal class.

The approach of Experiments 1 and 2 is unusual in that the goal is to use humans to investigate the behavior of algorithms. This makes sense because the algorithms are meant for recommending options to humans. In real-life contexts where recommendation is typically applied, however, there is no known ground truth, which makes assessing the performance of

⁵The concepts in Experiment 3 can be thought of as all the different decision boundaries a user may have.

algorithms difficult. One could assume a ground truth and perform computational simulations (e.g., Balabanović, 1998), but these assume that the simulation is robust to human-like variability, which is rarely known or checked. In these two experiments, humans were taught very simple concepts that governed relevance. They then labeled data for the algorithm, which captures the kinds of uncertainty associated with cognition—stochasticity across time in the responses to stimuli and in the representation of the concepts' features. The results bear the fruits of the approach. If one considers only the people who labeled correctly in the recommendation phase, active recommendation performs comparably well across a wide range of parameterizations. However, human variability is concentrated at the decision boundary and toward the non-focal concept, which gives parameterizations closer to pure recommendation a distinct advantage in recommendation and predictive accuracy.

Experiment 3 extended the approach of Experiments 1 and 2 to the domain of music recommendation, marking an important step towards bridging laboratory experiments to real-world recommendation. The calibration to the ground-truth relevance was removed to better match the typical ecological environment of recommendation. While this made the assessment of the algorithms' performance more difficult, Experiments 1 and 2 offered guidance on expected outcomes, which led to the crucial analysis that accounted for individuals' base rates. Compared to online experiments conducted by companies, Experiment 3 is unique in that potentially poor recommendations are incorporated in the design. Such recommendations allowed us to investigate the behavior of human and of human-algorithm interaction near the decision boundary and in the regime of anti-recommendation. The outcome is a complete, quantitative picture of the tradeoff between recommendation and prediction for a wide spectrum models spanning pure recommendation, active learning, and pure anti-recommendation.

Our proposed unified model of active recommendation takes pure recommendation and active learning as a starting point. Nevertheless, given a data-efficient classifier, we see that across a wide range of parameterizations the unified model exhibits behavior that is qualitatively different from either. That is, it achieves good performance on both goals of recommendation and active learning simultaneously. It will be useful to compare and contrast active recommendation with more explicit alternative approaches, namely, the ways that reinforcement learning manage the exploration-exploitation tradeoff. Of particular interests are ϵ -greedy algorithms, upper-confidence-bound algorithms, and Thompson sampling (Sutton and Barto, 2018; Russo et al., 2017). Such investigation may lead to reinterpretation of these well-known methods in the context of active recommendation by means of parameterization scheduling. It will also be useful to study collaborative filtering in the context of the unified model, as was done, albeit with semi-synthetic simulations, with pure recom-

mendation and active learning in (Sun et al., 2019), as this class of algorithms is amongst the highest performing methods in real-world recommendation.

Acknowledgements

This research was supported by NSF IIS and SBE NSF-1549981 to P.S and O.N. We thank Edwin Lee and Leon Johnson for contributing to the prototyping of Experiment 3.

Conflict of interest

We have no conflicts of interest to report.

References

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- Baeza-Yates, R. (2016). Data and algorithmic bias in the web. In *Proceedings of the 8th ACM Conference on Web Science*. ACM.
- Balabanović, M. (1998). Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction*, 8(1-2):71–102.
- Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. (2012). A contextual-bandit algorithm for mobile context-aware recommender system. In *International Conference on Neural Information Processing*, pages 324–331. Springer.
- Boutilier, C., Zemel, R. S., and Marlin, B. (2003). Active collaborative filtering. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 98–106.
- Bruner, J. S., Goodnow, J., and Austin, G. (1956). A study of thinking. Transaction Publishers, New Brunswick, New Jersey.
- Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. (2019). Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464.

- Coenen, A., Nelson, J. D., and Gureckis, T. M. (2019). Asking the right questions about the psychology of human inquiry: Nine open challenges. *Psychonomic Bulletin & Review*, 26(5):1548–1587.
- Crupi, V., Nelson, J. D., Meder, B., Cevolani, G., and Tentori, K. (2018). Generalized information theory meets human cognition: Introducing a unified framework to model uncertainty and information search. *Cognitive Science*, 42(5):1410–1456.
- Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.
- Elahi, M., Ricci, F., and Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.
- Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (2017). An algorithmic framework for performing collaborative filtering. In *ACM SIGIR Forum*, volume 51, pages 227–234. ACM New York, NY, USA.
- Houlsby, N., Huszar, F., Ghahramani, Z., and Hernández-Lobato, J. M. (2012). Collaborative gaussian processes for preference learning. In *Advances in neural information processing systems*, pages 2096–2104.
- Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.
- Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005.
- Liu, B., Wei, Y., Zhang, Y., Yan, Z., and Yang, Q. (2018). Transferable contextual bandit for cross-domain recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- MacKay, D. J. (1992). Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604.
- Maes, P. et al. (1994). Agents that reduce work and information overload. *Communications* of the ACM, 37(7):30–40.

- Markant, D. B. and Gureckis, T. M. (2014). Is it better to select or to receive? learning via active and passive hypothesis testing. *Journal of Experimental Psychology: General*, 143(1):94.
- Markant, D. B., Settles, B., and Gureckis, T. M. (2016). Self-directed learning favors local, rather than global, uncertainty. *Cognitive science*, 40(1):100–120.
- McInerney, J., Lacker, B., Hansen, S., Higley, K., Bouchard, H., Gruson, A., and Mehrotra, R. (2018). Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 31–39.
- Nasraoui, O. and Shafto, P. (2016). Human-algorithm interaction biases in the big data cycle: A markov chain iterated learning framework. arXiv preprint arXiv:1608.07895.
- Nelson, J. D. (2005). Finding useful questions: on bayesian diagnosticity, probability, impact, and information gain. *Psychological Review*, 112(4):979.
- Oaksford, M. and Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, 101(4):608.
- Pariser, E. (2011). The filter bubble: What the internet is hiding from you. *Penguin Press*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.
- Robertson, S. (1977). The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304.
- Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2017). A tutorial on thompson sampling. arXiv preprint arXiv:1707.02038.
- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.
- Salton, G., Fox, E. A., and Wu, H. (1983). Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036.
- Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

- Shepard, R. N., Hovland, C. I., and Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 75(13):1.
- Sparck Jones, K. (1970). Some thoughts on classification for retrieval. *Journal of Documentation*, 26(2):89–101.
- Sun, W., Khenissi, S., Nasraoui, O., and Shafto, P. (2019). Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 645–651, New York, NY, USA. Association for Computing Machinery.
- Sutton, R. S. and Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
- Szpektor, I., Maarek, Y., and Pelleg, D. (2013). When relevance is not enough: promoting diversity and freshness in personalized question recommendation. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1249–1260.
- Van Rijsbergen, C. (1979). Information Retrieval. Butterworths, London, UK.
- Wang, X., Wang, Y., Hsu, D., and Wang, Y. (2014). Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):1–22.
- Yang, L., Zhang, Y., Chen, J., Zhang, S., and Chen, D. Z. (2017). Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer.
- Yang, S. C.-H., Lengyel, M., and Wolpert, D. M. (2016). Active sensing in the categorization of visual patterns. *Elife*, 5:e12215.
- Zhang, H.-R., Min, F., He, X., and Xu, Y.-Y. (2015). A hybrid recommender system based on user-recommender interaction. *Mathematical Problems in Engineering*, 2015.